APPLICATION

FOR

UNITED STATES LETTERS PATENT

APPLICANT NAME R. G. Hartmann, et al.

TITLE SYSTEM AND METHOD FOR SERVER DISPLAY CONFIRMATION RECORD RESPONSE IN A CONNECTION ORIENTED CLIENT SERVER PROTOCOL

DOCKET NO. END9 2000 0172 US1

INTERNATIONAL BUSINESS MACHINES CORPORATION

CERTIFICATE OF MAILING UNDER 37 CFR 1 10

I hereby certify that, on the date shown below, this correspondence is being deposited with the United States Postal Service in an envelope addressed to the Commissioner of Patents and Trademarks, Washington, D C., 20231 as "Express Mail Post Office to Addressee" on 4/5/2001 Mailing Label No ET029404871US

Name of person mailing paper: Jennifer Smith

Jennifer Smith

a:formal2.lwp

SYSTEM AND METHOD FOR SERVER DISPLAY CONFIRMATION RECORD RESPONSE IN A CONNECTION ORIENTED CLIENT/SERVER PROTOCOL

Background of the Invention

Technical Field of the Invention

This invention pertains to connection oriented client/server negotiation protocols. More specifically, it pertains to Telnet negotiation protocols for display and printer sessions.

Background Art

There is a need in the art to enable a Telnet client when attempting to connect to a Telnet server to obtain connection status information including, for example, why did a connection request fail; why did a client auto-sign-on request fail; or what is the name of the virtual terminal display device assigned to this client. Auto-sign-on requests may fail, for example, because of an incorrect password or profile, a disabled or unknown profile, required encryption, expired user, and so forth.

This traditional Telnet support is accomplished in

accordance with the following suite of Network Working Group Request for Comments (RFCs): Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, May 1983; Postel, J. and J. Reynolds, "Telnet Option Specifications", STD 8, RFC 855, May 1983; Postel, J. and J. Reynolds, "Telnet Binary Transmission", STD 27, RFC 856, May 1983; VanBokkeln, J., "Telnet Terminal-Type Option", RFC 1091, February 1989; Postel, J. and J. Reynolds, "Telnet End of Record Option", RFC 885, December 1983; Alexander, S., "Telnet Environment Option", RFC 1572, January 1994; Chmielewski, P., "5250 Telnet Interface", RFC 1205, February 1991; Postel, J. and J. Reynolds, "Telnet Supress Go Ahead Option", STD 29, RFC 858, May 1983; and Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

The above suite of referenced RFCs jointly and severally fall short of providing an understanding of why a connection request has failed, and such is needed in the art to enable a client to correct the problem and retry a connection request such that it will be successful.

Similarly, when a connection request has succeeded, the client may need to know the name of the virtual terminal display device assigned to this client. Knowing the device

name of a client connection is useful for audit logging, billing and error analysis for connected clients.

Heretofore, screen scraping technology has been employed to acquire a device name, relying on the screen layout to analyze the location of the device name on the screen. If the sign-on panel is altered such that the device name is in a different location, screen scraping fails. Also, this screen scraping technology does not work when the sign-on panel is bypassed.

It is an object of the invention to provide an improved system and method for establishing a client/server connection.

It is a further object of the invention to provide an improved system and method for negotiating a client/server connection in a connection-oriented protocol.

It is a further object of the invention to provide a system and method for requesting and providing a confirmation record selectively including the virtual device name assigned by a server to a client device or an error code representing the cause of failure of connection.

It is a further object of the invention to provide a system and method for enabling a client to assign a session name to the GUI window for the client emulator responsive to a virtual device name assigned by a server to the client.

It is a further object of the invention to provide a system and method for providing to a client the device name assigned by a server to the client connection for audit logging, billing and error analysis.

Summary of the Invention

A system and method for operating a client to establish a network connection with a server. Environment parameters are negotiated for establishing a connection-oriented connection of the client to a server, the parameters including a request for the server to provide a confirmation record. Responsive to that request, the server provides the confirmation record to the client, the confirmation record selectively including the virtual device name assigned to the connection by the server or a return code indicative of a cause for failure to establish the connection.

In accordance with an aspect of the invention, there is provided a computer program product configured to be operable to operating a server in a network according to method steps including providing to a client a confirmation record including, for a successful connection, a virtual device name and, for an unsuccessful connection, a return code indicative of the cause of failure of the connection.

Other features and advantages of this invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

Figure 1 is a system diagram illustrating a client/server system.

Figure 2 is a diagram illustrating the format of a response record in accordance with the preferred embodiment of the invention.

Figure 3 is a flow chart representation of negotiations

for a confirmation record in accordance with the preferred embodiment of the invention.

Best Mode for Carrying Out the Invention

Referring to Figure 1, in accordance with the preferred embodiments of the invention, a confirmation record technology is provided for connection oriented client/server sessions, such as TCP/IP Telnet display sessions. This confirmation record technology is described hereafter and in T. Murphy, Jr., P. Rieth, J. Stevens, "5250 Telnet Enhancements", Network Working Group Request for Comments: 2877, July 2000, the teachings of which are incorporated by reference. With this technology, a Telnet client 40, for example, can connect to a Telnet server 42 over a network connection 44 and optionally request a detailed return code that describes the status of the connection. With the information of the return code, the client 40 is able to ascertain in the event of a successful connection the name of the virtual display device assigned to this client 40, and in the event of an unsuccessful connection the information required to correct the problem and retry a connection request such that it is successful. In the event

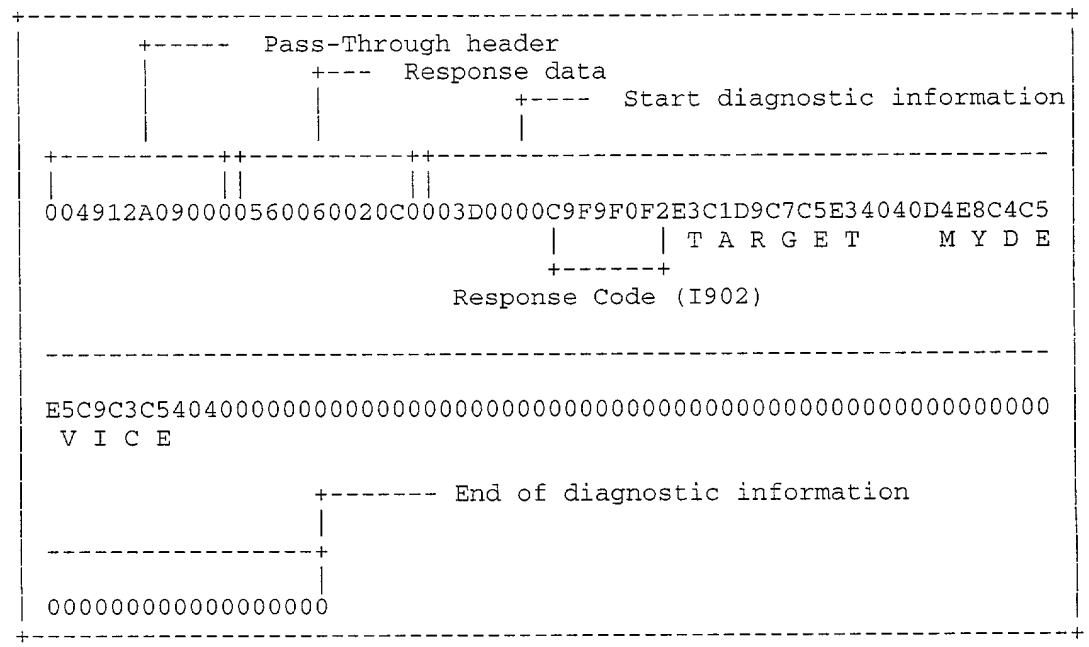
of a successful connection, the return code, or confirmation record, allows the client to know the virtual terminal device name without the need to employ a screen scrape scheme to analyze the sign-on panel, assuming it is even available. Knowing the virtual terminal device name enables the client to assign a session name to the GUI window for the client emulator. Also, knowing the device name of a client connection is very useful for audit logging, billing and error analysis for connection clients.

Referring to Figure 2, the format of a response record 100 includes pass through header 102, response data 104, and diagnostic information 106. Pass through header 108 includes length field 108, header 110, and several characters from fixed value fields 112. Response data 104 includes several characters from field 112. Diagnostic information includes a few characters from field 112, response code 114, system name 118 and device name 120.

In accordance with a preferred embodiment of the invention, Table 1 presents an example of a success response record 100 according to the format of Figure 2, and Table 2 presents an error response record 100 according to the same format. Table 3 gives some of the response codes 114 for a

success response 100 and Table 4 some of the response codes 114 for an error response record 100. The response record in Table 2 is one that reports an error. In this example , the virtual device named "MYDEVICE", is not available on the target system "TARGET", because the device is not available. This error may indicate that the device was already assigned to another Telnet session.

Table 1 Example Success Response Record



- '0049'X = Length pass-through data, including this length field
- '12A0'X = GDS LU6.2 header
- '900005600600020C0003D0000'X = Fixed value fields
- 'C9F9F0F2'X = Response Code (I902)
- 'E3C1D9C7C5E34040'X = System Name (TARGET)
- 'D4E8C4C5E5C9C3C54040'X = Object Name (MYDEVICE)

Table 2 Example Error Response Record

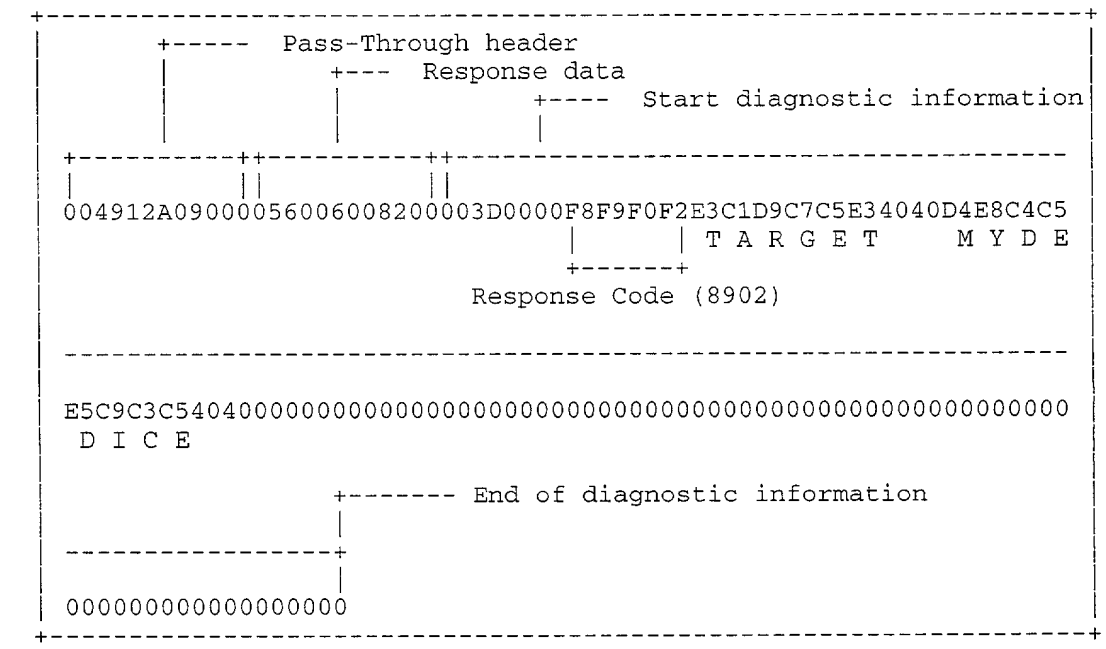


Figure 2. Example of an error response record.

- '0049'X = Length pass-through data, including this length field
- '12A0'X = GDS LU6.2 header
- '90000560060020C0003D0000'X = Fixed value fields
- 'F8F9F0F2'X = Response Code (8902)
- 'E3C1D9C7C5E34040'X = System Name (TARGET)
- 'D4E8C4C5E5C9C3C54040'X = Object Name (MYDEVICE)

Table 3 Start-Up Response Record Success Response Codes

CODE	DESCRIPTION
I901	Virtual device has less function than source device
I902	Session successfully started
I906	Automatic sign-on requested, but not allowed. Session still allowed; a sign-on screen will be coming.

Table 4 Start-Up Response Record Error Response Codes

CODE	DESCRIPTION
2702	Device description not found.
2703	Controller description not found.
2777	Damaged device description.
8901	Device not varied on.
8902	Device not available.
8903	Device not valid for session.
8906	Session initiation failed.
8907	Session failure.
8910	Controller not valid for session.
8916	No matching device found.
8917	Not authorized to object.
8918	Job canceled.
8920	Object partially damaged.
8921	Communications error.
8922	Negative response received.
8923	Start-up record built incorrectly.
8925	Creation of device failed.
8928	Change of device failed.
8929	Vary on or vary off failed.
8930	Message queue does not exist.
8934	Start-up for S/36 WSF received.
8935	Session rejected.
8936	Security failure on session attempt.
8937	Automatic sign-on rejected.
8940	Automatic configuration failed or not allowed.
I904	Source system at incompatible release.

Referring to Figure 3, method steps of an exemplary negotiation for a confirmation record are summarized in accordance with the preferred embodiment of the invention.

In step 50, server 42 invites client 40 to engage in new environment negotiations. These negotiations are conducted in accordance with procedures described in S. Alexander, "Telnet Environment Options Negotiations", RFC 1572, Jan. 1994.

In step 52, client 40 accepts the invitation to negotiate a new environment.

In step 54, server 42 opens negotiations for terminal type, which client 40 accepts in step 56.

In step 58, server 42 instructs client 40 to send several parameters, and in step 60 client 40 responds. In accordance with the preferred embodiment of the invention, in the response of step 60, client 40 requests with the code "USERVAR 'IBMSENDCONFREC' VALUE 'YES'" that server 42 send a confirmation record 100. Alternatively, such a request may be implied from some other parameter in connection with the new environment negotiations. Thus, for example, client 40 may have to specifically request a confirmation record 100 when requesting connection of a virtual display device, but such would be implied when requesting connection of a virtual printer device.

Negotiations continue, for such additional environment parameters as end-of-record and binary, and then in step 66 server 42 transmits the confirmation record, followed in step 68 in this example of a successful connection with the data stream.

In Table 5, an expanded example is presented of environment option negotiations similar to those of Figure 3. As shown, clear text is followed by hex representation. Thus, line 2 'FFFD27' is the hex representation of line 1 'IAC DO NEW-ENVIRON', lines 13-14 are the hex representation of lines 9-12, and lines 58-62 are a hex representation of the confirmation record of Figure 2. The request for a confirmation record is illustrated at line 24. In line 59, the hex value 'C9F9F0F2' represents the successful return code 114 of I902 (see Table 3), and the device name 120 assigned to this virtual device is in the following ten hex bytes 'D1C5C6C6 E2C4E2D7 4040' on lines 59 and 60. IAC is a Telnet option negotiation code meaning "Interpret as command", SB represents "begin" and SE "end".

Table 5: TN5250E Environment Option Negotiations

	Telnet Server		Telnet Client
1	IAC DO NEW-ENVIRON	->	
2	FFFD27		
3		<-	IAC WILL NEW-ENVIRON
4			FFFB27
5	IAC DO TERMTYPE	->	
6	FFFD18		
7		<-	IAC WILL TERMTYPE
8			FFFB18
9	IAC SB NEW-ENVIRON SEND		
10	USERVAR "IBMRSEEDxxxxxxxx"		
11	USERVAR "IBMSUBSPW"		
12	VAR USERVAR IAC SE	->	
13	FFFA2701 0349424D 52534545		
14	447D68B9 2BE04E04 040003FF F0		
15			IAC SB NEW-ENVIRON IS
16			VAR "USER" VALUE "JSTEVENS"
17			USERVAR "IBMRSEED" VALUE
18			USERVAR "IBMSUBSPW" VALUE
19			"YYYYYYYY"
20			USERVAR "DEVNAME" VALUE "JEFFSDSP"
21			USERVAR "CODEPAGE" VALUE "37"
22			USERVAR "CHARSET" VALUE "697"
23			USERVAR "KBDTYPE" VALUE "USB"
24			USERVAR "IBMSENDCONFREF" VALUE "YES"
25		<-	IAC SE
26			FFFA2700 00555345 52014A53 54455645
27			4E530349 424D5253 45454401 04696CD0
28			D7C41F81 0349424D 53554253 50570131
29			96A30203 3F5321FD 03444556 4E414D45
30			014A4546 46534453 5003434F 44455041
31			47450133 37034348 41525345 54013639
32			37034B42 44545950 45015553 4249424D
33			53454E44 434F4E46 52454301 594553FF
34			F0
35			
36	IAC SB TERMTYPE SEND		
37	IAC SE	->	
38	FFFA1801 FFF0		
39		<-	IAC SB TERMTYPE IS IBM-3179-2 IAC SE
40			FFFA1800 49424D2D 33313739 2D32FFF0
41	IAC DO EOR	->	
42	FFFD19		
43		<-	IAC WILL EOR
44			FFFB19
45	IAC WILL EOR	->	
46	FFFB19		
47		<-	IAC DO EOR
48			FFFD19
49	IAC DO BINARY	->	
50	FFFD00		
51		<-	IAC WILL BINARY

```
                                          FFFB00
IAC WILL BINARY                    ->
FFFB00
                                   <- IAC DO BINARY
                                      FFFD00
Display Confirmation Record        ->
004912A0 90000560 060020C0 003D0000
C9F9F0F2 D9E2F0F1 F0404040 D1C5C6C6
E2C4E2D7 40400000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00FFEF

RFC 1205 Data Stream               ->
001112A0 00000400 000304F3 0005D970
00FFEF
```

Device name collision occurs when a Telnet client 40 sends the Telnet server 42 a virtual device name that it wants to use, but that device is already in use on the server 42. When this occurs, the Telnet server 42 sends a request to the client 40 asking it to try another device name. The environment option negotiation uses the USERVAR name of DEVNAME to communicate the virtual device name. Table 6 shows how the Telnet server 42 requests the Telnet client 40 to send a different DEVNAME when device name collision occurs, and is an example of how negotiations are done using environment variables, such as DEVNAME, USER, CODEPAGE, CHARSET, and so forth. These are negotiations for various display session attributes which, according to the present invention, is enhanced to include IBMSENDCONFREC.

Table 6 Negotiating Display Session Attributes

AS/400 Telnet server		Enhanced Telnet client
IAC SB NEW-ENVIRON SEND VAR USERVAR IAC SE	-->	
Server requests all environment variables be sent.		
		IAC SB NEW-ENVIRON IS USERVAR "DEVNAME" VALUE "MYDEVICE1" USERVAR "xxxxx" VALUE "xxx" ... IAC SE
	<--	
Client sends all environment variables, including DEVNAME. Server tries to select device MYDEVICE1. If the device is already in use, server requests DEVNAME be sent again.		
IAC SB NEW-ENVIRON SEND USERVAR "DEVNAME" IAC SE	-->	
Server sends a request for a single environment variable: DEVNAME		
		IAC SB NEW-ENVIRON IS USERVAR "DEVNAME" VALUE "MYDEVICE2" IAC SE
	<--	
Client sends one environment variable, calculating a new value of MYDEVICE2. If MYDEVICE2 is different from the last request, then server tries to select device MYDEVICE2, else server disconnects client. If MYDEVICE2 is also in use, server will send DEVNAME request again, and keep doing so until it receives a device that is not in use, or the same device name twice in row.		

Advantages over the Prior Art

It is an advantage of the invention that there is provided an improved system and method for establishing a client/server connection.

It is a further advantage of the invention that there

is provided an improved system and method for negotiating a client/server connection in a connection-oriented protocol.

It is a further advantage of the invention that there is provided a system and method for requesting and providing a confirmation record selectively including the virtual device name assigned by a server to a client device or an error code representing the cause of failure of connection.

It is a further advantage of the invention that there is provided a system and method for enabling a client to assign a session name to the GUI window for the client emulator responsive to a virtual device name assigned by a server to the client.

It is a further advantage of the invention that there is provided a system and method for providing to a client the device name assigned by a server to the client connection for audit logging, billing and error analysis.

Alternative Embodiments

It will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. In particular, it is within the scope of the invention to provide a computer program product or program element, or a program storage or memory device such as a solid or fluid transmission medium, magnetic or optical wire, tape or disc, or the like, for storing signals readable by a machine, for controlling the operation of a computer according to the method of the invention and/or to structure its components in accordance with the system of the invention.

Further, each step of the method may be executed on any general computer, such as an IBM System 390 (z Series), AS/400 (I Series), PC (x Series), p Series, or the like and pursuant to one or more, or a part of one or more, program elements, modules or objects generated from any programming language, such as C++, Java, Pl/1, Fortran or the like. And still further, each said step, or a file or object or the like implementing each said step, may be executed by special

purpose hardware or a circuit module designed for that purpose.

While the preferred embodiment of the invention has been described primarily with respect to a Telnet environment or protocol, in a broader sense it is applicable to any connection oriented client/server protocol, such as a TCP/IP family of applications. Such protocols may make use of a confirmation record, served in accordance with the preferred embodiments of the present invention, confirming the status or other attributes associated with an actual connection. An example of such a protocol is the file transfer protocol (FTP), in which a connection is initiated and held for the duration of a file transfer. Telnet initiates and holds the connection for the duration of the dialogue between the attaching client emulator that initiates the connection to a targeted host server and its application.

Accordingly, the scope of protection of this invention is limited only by the following claims and their equivalents.